

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.

THIS PAGE BLANK (USPTO)

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

19.04.99

EJ

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

1998年 7月 3日

REC'D 14 JUN 1999	
WIPO	PCT

出 願 番 号
Application Number:

平成10年特許願第189516号

出 願 人
Applicant(s):

大見 忠弘

PRIORITY
DOCUMENT

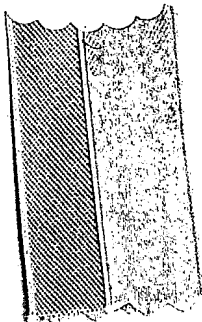
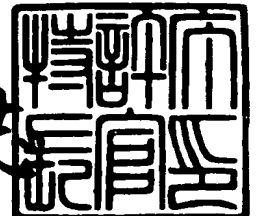
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)



1999年 5月28日

特許庁長官
Commissioner,
Patent Office

伴佐山 建志



出証番号 出証特平11-3033339

【書類名】 特許願

【整理番号】 XY10343

【提出日】 平成10年 7月 3日

【あて先】 特許庁長官殿

【国際特許分類】 H01L 21/00

【発明の名称】 ベクトル量子化で用いるコードブックの作成装置および方法、ベクトル量子化装置、記録媒体

【請求項の数】 21

【発明者】

 【住所又は居所】 宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内

 【氏名】 中山 貴裕

【発明者】

 【住所又は居所】 宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内

 【氏名】 森本 達郎

【発明者】

 【住所又は居所】 宮城県仙台市青葉区米ヶ袋2の1の17の301

 【氏名】 大見 忠弘

【特許出願人】

 【識別番号】 000205041

 【氏名又は名称】 大見 忠弘

【代理人】

 【識別番号】 100090273

 【弁理士】

 【氏名又は名称】 國分 孝悦

 【電話番号】 03-3590-8901

【手数料の表示】

 【予納台帳番号】 035493

 【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ベクトル量子化で用いるコードブックの作成装置および方法、
ベクトル量子化装置、記録媒体

【特許請求の範囲】

【請求項1】 少なくとも1つ以上のデータを有するデータ列であるベクトルの集合から成り、ベクトル量子化で用いられるコードブックを作成する装置において、

上記ベクトルを構成するブロック内でデータ値が徐々に変化していくベタパターンコードを少なくとも1種類作成するベタパターン作成手段と、

上記ブロック内でデータ値が急激に変化するエッジパターンコードを少なくとも1種類作成するエッジパターン作成手段とを備えたことを特徴とするベクトル量子化で用いるコードブックの作成装置。

【請求項2】 上記ベタパターン作成手段およびエッジパターン作成手段により作成された各々のパターンコードを記憶する記憶手段と、

上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のパターンコードに対して演算を施すことにより、これらとは異なるパターンコードを作成するコード演算手段とを備えたことを特徴とする請求項1に記載のベクトル量子化で用いるコードブックの作成装置。

【請求項3】 上記ベタパターン作成手段は、上記ブロック内の任意の辺および任意の角を始点として対向する辺および角に向かってデータ値が徐々に変化していくパターンコードを作成することを特徴とする請求項1または2に記載のベクトル量子化で用いるコードブックの作成装置。

【請求項4】 上記ベタパターン作成手段は、上記ブロック内の全てのデータ値が、とり得る範囲の中間値よりも小さいかあるいは大きい値で構成されるパターンコードを作成することを特徴とする請求項1～3の何れか1項に記載のベクトル量子化で用いるコードブックの作成装置。

【請求項5】 上記ベタパターン作成手段は、徐々に変化する度合いが同じでデータ値自体が異なるパターンコードを複数作成することを特徴とする請求項1～4の何れか1項に記載のベクトル量子化で用いるコードブックの作成装置。

【請求項 6】 上記エッジパターン作成手段は、基準のパターンを入力するパターン入力手段と、

上記入力された基準のパターンを、あらかじめ決められたデータ値群のみで表すようにする量子化手段とを備えることを特徴とする請求項 1 または 2 に記載のベクトル量子化で用いるコードブックの作成装置。

【請求項 7】 上記エッジパターン作成手段は、上記ブロック内のデータ値間の差が同じでデータ値自体が異なるパターンコードを複数作成することを特徴とする請求項 1、2 または 6 に記載のベクトル量子化で用いるコードブックの作成装置。

【請求項 8】 上記エッジパターン作成手段は、上記ブロック内のデータ値間の差が同じものに加えて、差の比率が同じでデータ値自体が異なるパターンコードを複数作成することを特徴とする請求項 7 に記載のベクトル量子化で用いるコードブックの作成装置。

【請求項 9】 上記コード演算手段は、上記記憶手段に記憶されている各々のパターンコードに対して、ブロックの回転処理およびデータ値の反転処理の少なくとも何れか一方の演算を施すことを特徴とする請求項 2 に記載のベクトル量子化で用いるコードブックの作成装置。

【請求項 10】 少なくとも 1 つ以上のデータを有するデータ列であるベクトルの集合から成り、ベクトル量子化で用いられるコードブックを作成する方法において、

上記ベクトルを構成するブロック内でデータ値が徐々に変化していく少なくとも 1 種類のベタパターンコードと、上記ブロック内でデータ値が急激に変化する少なくとも 1 種類のエッジパターンコードとをあらかじめ基本パターンとして作成して記憶手段に記憶しておき、

上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のパターンコードに対して演算を施すことにより、上記基本パターンとは異なるパターンコードを作成するようにしたことを特徴とするベクトル量子化で用いるコードブックの作成方法。

【請求項 11】 上記ベクトル量子化の実行に先立って、上記ブロック内の

任意の辺を始点として対向する辺に向かってデータ値が徐々に変化していくベタパターンコード、および上記ブロック内の任意の角を始点として対向する角に向かってデータ値が徐々に変化していくベタパターンコードを上記基本パターンとして作成し、上記記憶手段に記憶しておくことを特徴とする請求項 10 に記載のベクトル量子化で用いるコードブックの作成方法。

【請求項 12】 上記ベクトル量子化の実行に先立って、基準のパターンを入力し、入力した基準のパターンをあらかじめ決められたデータ値のみで表すようにする量子化処理を少なくとも行うことによって上記エッジパターンコードを作成し、上記基本パターンとして上記記憶手段に記憶しておくことを特徴とする請求項 10 に記載のベクトル量子化で用いるコードブックの作成方法。

【請求項 13】 上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のパターンコードに対して、ブロックの回転処理およびデータ値の反転処理の少なくとも何れか一方の演算を施すことを特徴とする請求項 10 に記載のベクトル量子化で用いるコードブックの作成方法。

【請求項 14】 少なくとも 1 つ以上のデータを有するデータ列をブロック化してベクトルと成し、あらかじめ用意されたコードブックの中から、圧縮対象より抽出されるベクトルに類似したコードベクトルを探し出して、それに対応するコードを出力するベクトル量子化装置において、

上記ブロック内でデータ値が徐々に変化していくベタパターンコードを少なくとも 1 種類作成するベタパターン作成手段と、

上記ブロック内でデータ値が急激に変化するエッジパターンコードを少なくとも 1 種類作成するエッジパターン作成手段と、

上記ベタパターン作成手段およびエッジパターン作成手段により作成された各々のパターンコードを記憶する記憶手段と、

上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のパターンコードに対して演算を施すことにより、これらとは異なるパターンコードを作成するコード演算手段と、

上記記憶手段から読み出されたパターンコードおよび上記コード演算手段により作成されたパターンコードを利用して上記ベクトル量子化を実行するベクトル

量子化手段とを備えたことを特徴とするベクトル量子化装置。

【請求項 15】 少なくとも 1 つ以上のデータを有するデータ列であるベクトルの集合から成り、ベクトル量子化で用いられるコードブックを作成する方法において、

少なくとも 1 種類のパターンコードをあらかじめ基本パターンとして作成して記憶手段に記憶しておき、

上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている上記少なくとも 1 種類のパターンコードに対して演算を施すことにより、上記基本パターンとは異なるパターンコードを作成するようにしたことを特徴とするベクトル量子化で用いるコードブックの作成方法。

【請求項 16】 上記演算は、回転処理およびデータ値の反転処理の少なくとも何れか一方であることを特徴とする請求項 15 に記載のベクトル量子化で用いるコードブックの作成方法。

【請求項 17】 請求項 15 または 16 に記載のコードブック作成方法の処理手順をコンピュータに実行させるためのプログラムを記録したことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 18】 少なくとも 1 つ以上のデータを有するデータ列をブロック化してベクトルと成し、あらかじめ用意されたコードブックの中から、圧縮対象より抽出されるベクトルに類似したコードベクトルを探し出して、それに対応するコードを出力するベクトル量子化装置において、

上記コードブックを記憶する記憶手段と、

上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のコードに対して演算を施すことにより、これらとは異なるコードを作成する演算手段と、

上記記憶手段から読み出されたコードおよび上記演算手段により作成されたコードを利用して上記ベクトル量子化を実行するベクトル量子化手段とを備えたことを特徴とするベクトル量子化装置。

【請求項 19】 上記演算手段による演算は、回転処理およびデータ値の反転処理の少なくとも何れか一方であることを特徴とする請求項 18 に記載のベク

トル量子化で用いるベクトル量子化装置。

【請求項20】 少なくとも1つ以上のデータを有するデータ列をブロック化してベクトルと成し、あらかじめ用意されたコードブックの中から、圧縮対象より抽出されるベクトルに類似したコードベクトルを探し出して、それに対応するコードを出力するベクトル量子化プログラムを記録した記録媒体であって、

上記ベクトル量子化の実行の際に、記憶手段に記憶されている各々のコードに対して演算を施すことにより、これらとは異なるコードを作成する演算ステップと、

上記記憶手段から読み出されたコードおよび上記演算ステップにより作成されたコードを利用して上記ベクトル量子化を実行するベクトル量子化ステップとをコンピュータに実行させるためのプログラムを記録したことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項21】 上記演算ステップによる演算として、回転処理およびデータ値の反転処理の少なくとも何れか一方を行うようにしたことを特徴とする請求項20に記載のコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ベクトル量子化で用いるコードブックの作成装置および方法、ベクトル量子化装置、更にはこれらの処理を行うためのプログラムを記憶した記録媒体に関するものである。

【0002】

【従来の技術】

従来、データ圧縮の手法が種々提案されている。その中で、圧縮データの伸長処理を非常に簡単に行うことが可能なデータ圧縮アルゴリズムの1つとして、「ベクトル量子化」という手法が良く知られている。このアルゴリズムは、古くから信号処理の分野で知られており、特に、画像信号や音声信号のデータ圧縮、あるいはパターン認識に応用されてきた。

【0003】

このベクトル量子化では、ある大きさ（例えば4×4画素のブロック）の画素パターン（コード）を幾つか用意しておき、それぞれにユニークな番号などを与える（この集合体を「コードブック」という）。そして、例えば2次元配列の画像データ中から同じ大きさ（例えば4×4画素）のブロックを順次取り出し、それと最も似通ったパターンをコードブック中から見つけ出して、そのパターンの番号を当該ブロックに当てはめるというデータ圧縮を行う。ベクトル量子化では、1つのブロック内のデータ列が1つのベクトルに対応する。

【0004】

このようにコード化された圧縮データの受信側あるいは伸長側では、各ブロック毎に番号に対応するパターンをコードブックの中から取り出すだけで、元の画像を再現することができる。したがって、伸長側では、コードブックさえ受け取っているか、あるいはあらかじめ保持していれば、特に特殊な演算は必要としないため、非常に簡単なハードウェアで元の画像を再生することが可能となる。

【0005】

【発明が解決しようとする課題】

上述のようなベクトル量子化を実行する上で、必ず必要となるのがコードブックの作成である。そして、ベクトル量子化の特性上、再現される再生画像の良否は、使用するコードブックの良否と密接に関係している。したがって、例えば画像のデータ圧縮を行う際に、高い圧縮率を保持したまま高画質の再生画像を得るようになるためには、このコードブックとしていかに性能の良いものを作成するかが課題となっている。

【0006】

従来、コードブックの最適化の手法としては、Kohonenの自己組織化マップの手法などを始めとして幾つかの手法が知られている。これらの手法では、サンプル画像などを用いて適当な数式処理を行うことにより、コードブックの最適化を図るものである。しかしながら、従来の最適化技術は何れも、得られるコードブックは、最適化の際に使用したデータに対してのみ有用なコードブックになってしまうという問題があった。

【0007】

すなわち、例えば、ある人の顔の画像データを用いて最適化されたコードブックは、その最適化に用いた画像に対しては最良のコードブックとなるが、他の画像に対しては必ずしも最良のコードブックになるとは限らない。したがって、例えば、そのコードブックを他の人の顔の画像データに対して用いてデータ圧縮を実施すると、圧縮データから再生した画像の画質は低下することになる。

【0008】

さらに、最適化に用いた画像と同じ人の顔という分類に含まれる画像に対しては、再生画像として比較的良好な画質が得られても、風景や文字といった異なる分類の画像に対しては、画質が劣化してしまうことが多い。つまり、コードブックに含まれているパターンが画像によって全く異なっているため、汎用性の低いコードブックになってしまうという問題があった。

【0009】

そこで、どのような分類の画像を圧縮した場合でも、再生画像の良好な画質を得るために、様々なサンプル画像を用いて最適化を行い、これにより得られるコードブックを全て登録しておくということが考えられる。このようにすれば、実際にベクトル量子化を実施する際には、人物、風景、文字といった様々な分類の画像に合ったコードブックがあらかじめ用意されているので、その中から元画像に近いパターンを選び出すことが可能となる。

【0010】

しかしながら、この場合は、あらかじめ用意しておくべきコード数が膨大なものとなり、これを記憶しておくためのメモリの容量が非常に大きくなってしまいう問題があった。メモリの容量が大きくなると、装置全体の規模も大きくなってしまい、小型化が困難な状況となる。また、コストもそれだけ増大する結果となってしまいうため、このような手法を採用するのは現実的でない。

【0011】

本発明は、このような問題を解決するために成されたものであり、種々の画像に対応できる汎用性の高いコードブックを実現できるようにすることを第1の目的とする。

また、本発明は、コードブックを記憶しておくためのメモリ容量を増大させる

ことなく汎用性の高いコードブックを実現できるようにすることを第2の目的とする。

【0012】

【課題を解決するための手段】

本発明のベクトル量子化で用いるコードブックの作成装置は、少なくとも1つ以上のデータを有するデータ列であるベクトルの集合から成り、ベクトル量子化で用いられるコードブックを作成する装置において、上記ベクトルを構成するブロック内でデータ値が徐々に変化していくベタパターンコードを少なくとも1種類作成するベタパターン作成手段と、上記ブロック内でデータ値が急激に変化するエッジパターンコードを少なくとも1種類作成するエッジパターン作成手段とを備えたことを特徴とする。

【0013】

本発明の他の態様では、上記ベタパターン作成手段およびエッジパターン作成手段により作成された各々のパターンコードを記憶する記憶手段と、上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のパターンコードに対して演算を施すことにより、これらとは異なるパターンコードを作成するコード演算手段とを備えたことを特徴とする。

【0014】

また、本発明のベクトル量子化で用いるコードブックの作成方法は、少なくとも1つ以上のデータを有するデータ列であるベクトルの集合から成り、ベクトル量子化で用いられるコードブックを作成する方法において、上記ベクトルを構成するブロック内でデータ値が徐々に変化していく少なくとも1種類のベタパターンコードと、上記ブロック内でデータ値が急激に変化する少なくとも1種類のエッジパターンコードとをあらかじめ基本パターンとして作成して記憶手段に記憶しておき、上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のパターンコードに対して演算を施すことにより、上記基本パターンとは異なるパターンコードを作成するようにしたことを特徴とする。

【0015】

本発明の他の態様では、少なくとも1つ以上のデータを有するデータ列である

ベクトルの集合から成り、ベクトル量子化で用いられるコードブックを作成する方法において、少なくとも1種類のパターンコードをあらかじめ基本パターンとして作成して記憶手段に記憶しておき、上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている上記少なくとも1種類のパターンコードに対して演算を施すことにより、上記基本パターンとは異なるパターンコードを作成するようにしたことを特徴とする。

【0016】

また、本発明のベクトル量子化装置は、少なくとも1つ以上のデータを有するデータ列をブロック化してベクトルと成し、あらかじめ用意されたコードブックの中から、圧縮対象より抽出されるベクトルに類似したコードベクトルを探し出して、それに対応するコードを出力するベクトル量子化装置において、上記ブロック内でデータ値が徐々に変化していくベタパターンコードを少なくとも1種類作成するベタパターン作成手段と、上記ブロック内でデータ値が急激に変化するエッジパターンコードを少なくとも1種類作成するエッジパターン作成手段と、上記ベタパターン作成手段およびエッジパターン作成手段により作成された各々のパターンコードを記憶する記憶手段と、上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のパターンコードに対して演算を施すことにより、これらとは異なるパターンコードを作成するコード演算手段と、上記記憶手段から読み出されたパターンコードおよび上記コード演算手段により作成されたパターンコードを利用して上記ベクトル量子化を実行するベクトル量子化手段とを備えたことを特徴とする。

【0017】

本発明の他の態様では、少なくとも1つ以上のデータを有するデータ列をブロック化してベクトルと成し、あらかじめ用意されたコードブックの中から、圧縮対象より抽出されるベクトルに類似したコードベクトルを探し出して、それに対応するコードを出力するベクトル量子化装置において、上記コードブックを記憶する記憶手段と、上記ベクトル量子化の実行の際に、上記記憶手段に記憶されている各々のコードに対して演算を施すことにより、これらとは異なるコードを作成する演算手段と、上記記憶手段から読み出されたコードおよび上記演算手段に

より作成されたコードを利用して上記ベクトル量子化を実行するベクトル量子化手段とを備えたことを特徴とする。

【0018】

また、本発明のコンピュータ読み取り可能な記録媒体は、請求項15または16に記載のコードブック作成方法の処理手順をコンピュータに実行させるためのプログラムを記録したことを特徴とする。

【0019】

本発明の他の態様では、少なくとも1つ以上のデータを有するデータ列をブロック化してベクトルと成し、あらかじめ用意されたコードブックの中から、圧縮対象より抽出されるベクトルに類似したコードベクトルを探し出して、それに対応するコードを出力するベクトル量子化プログラムを記録した記録媒体であって、上記ベクトル量子化の実行の際に、記憶手段に記憶されている各々のコードに対して演算を施すことにより、これらとは異なるコードを作成する演算ステップと、上記記憶手段から読み出されたコードおよび上記演算ステップにより作成されたコードを利用して上記ベクトル量子化を実行するベクトル量子化ステップとをコンピュータに実行させるためのプログラムを記録したことを特徴とする。

【0020】

【発明の実施の形態】

以下、本発明の一実施形態を図面に基づいて説明する。

図1は、本発明の一実施形態であるコードブック作成装置の構成例を示す機能ブロック図、図2は、作成されたコードブックを利用してデータ圧縮を実行するベクトル量子化装置の構成例を示す機能ブロック図である。また、図3および図4は、作成されるコードベクトル（パターン画像）の例を示す図である。

【0021】

一般に、顔画像は画素値が全体的に滑らかに変化するパターンが大半を占めており、その変化は単調で、変化量は非常に小さいものである。また、風景画像などにおいても、部分的に見ると、顔画像と同じように画素値が全体的に滑らかに変化するパターンが存在する。一方、文字などにおいては、画素値の変化が急激で、変化量が非常に大きいのが一般である。また、風景画像などの中にも、文字

と同じように画素値が急激に変化する部分が存在する。

【0022】

このように、様々な分類に属するあらゆる画像は、大きく分けると、画素値が全体的に滑らかに変化するパターン（以下、これを「ベタパターン」と称する）の部分と、画素値が急激に変化するパターン（以下、これを「エッジパターン」と称する）の部分とから構成されることが分かった。そこで、本実施形態では、この画像の性質に着目し、ベタパターン用のコードとエッジパターン用のコードとを作成するようにした。これら2つのパターンのコードをうまく組み合わせることで、種々の画像に対応することが可能となる。

【0023】

ベタパターンのコードとしては、例えば図3に示すように、8つの方向に単調に変化するパターンを作成することとした。すなわち、4×4画素単位で構成されるブロックのエッジ部分（上下左右の各辺および四隅の各点）の何れかを始点として、画素値（例えば輝度値）が徐々に変化するパターンを作成する。なお、ここでは一例として4×4画素単位のコードベクトルを作成しているが、この大きさに限定されるものではない。

【0024】

図3において、（a）はブロックの左辺を始点として、縦1列を1単位として輝度値が右辺に向かってA→B→C→Dのように徐々に大きくなっていくパターン、（b）はブロックの右辺を始点として、縦1列を1単位として輝度値が左辺に向かって徐々に大きくなっていくパターン、（c）はブロックの下辺を始点として、横1列を1単位として輝度値が上辺に向かって徐々に大きくなっていくパターン、（d）はブロックの上辺を始点として、横1列を1単位として輝度値が下辺に向かって徐々に大きくなっていくパターンである。

【0025】

また、（e）はブロックの左上角を始点として、斜め1列を1単位として輝度値が右下角に向かって徐々に大きくなっていくパターン、（f）はブロックの右上角を始点として、斜め1列を1単位として輝度値が左下角に向かって徐々に大きくなっていくパターン、（g）はブロックの左下角を始点として、斜め1列を

1 単位として輝度値が右上角に向かって徐々に大きくなっていくパターン、(h) はブロックの右下角を始点として、斜め 1 列を 1 単位として輝度値が左上角に向かって徐々に大きくなっていくパターンである。

【0026】

図 3 に示したこれら 8 種類のパターンのうち、ベクトル量子化を実際に実行するに先立ってあらかじめ作成して登録しておくものは、図 3 (a) ~ (d) の中の何れか 1 つと、図 3 (e) ~ (h) の中の何れか 1 つの合計 2 種類である。残りの 6 種類のパターンは、後述するように、ベクトル量子化を実際に実行する際に作成される。なお、以下では説明の都合上、図 3 (a) と (e) のパターン（以下、これを「基本パターン」と称する）をあらかじめ作成するものとする。

【0027】

ここで、ベタの基本パターンを作成する手順を、図 1 に示したコードブック作成装置および図 5 に示すフローチャートを用いて説明する。

図 5 において、まずステップ S 1 で、例えば 4 × 4 画素単位のブロックを幾つかのグループに分ける。例えば、図 3 (a) の基本パターンの場合は、縦 1 列を 1 つのグループとして A ~ D の 4 つのグループに分ける。

【0028】

このグループ分けを行う際には、まずユーザが図 1 の始点指定部 1 を用いて、ブロック内のどこを始点にするかを指定する。図 3 (a) のパターンを基本パターンとして登録する場合は、ブロック内の左辺 4 つの画素（グループ A）を始点として指定する。なお、この始点指定部 1 は、例えばキーボードやマウス等の入力デバイスによって構成される。

【0029】

グループ化部 2 は、始点指定部 1 によってどこが始点として指定されたかに応じて、グループ分けを行う。例えば、図 3 (a) や (b) を基本パターンとする場合は、縦方向の 1 列を 1 つのグループとしてグループ分けを行い、図 3 (c) や (d) を基本パターンとする場合は、横方向の 1 列を 1 つのグループとしてグループ分けを行う。また、図 3 (e) ~ (h) を基本パターンとする場合は、斜め方向の 1 列を 1 つのグループとしてグループ分けを行う。

【0030】

なお、上述したように、図3（a）～（d）のパターン、（e）～（h）のパターンは、それぞれの中で1つを基本パターンとして登録しておけば、残りは全て演算によって作成される。したがって、例えば、基本パターンは必ず図3（a）および（e）のパターンであると固定しても良い。この場合は、始点指定部1は不要である。

【0031】

次に、ステップS2でユーザは、図1の増分入力部3を用いて、上記ブロック内の始点から見た終点の輝度値の増分hを少なくとも1つ入力する。なお、この増分入力部3も、例えばキーボードやマウス等の入力デバイスによって構成される。また、この増分hは、ユーザが入力するのではなく、あらかじめ決められたデフォルト値を装置が設定するようにしても良い。

【0032】

次に、ステップS3では、図1の始点輝度値設定部4により、上記ブロック内の始点の輝度値を設定する。ここで始点の輝度値を与えるときは、例えば、

0～（輝度値としてとり得る範囲の中間値－増分h）

の間をn等分し、それぞれの等分値を始点の輝度値として夫々設定する。なお、ここでは、等分値を与えているが、上記0～（輝度値としてとり得る範囲の中間値－増分h）の範囲内であれば、必ずしも等分値である必要はない。また、ここでは演算によって複数の輝度値を与えているが、上記の範囲内でユーザが自分の判断で始点の輝度値を任意に入力するようにしても良い。

【0033】

このようにして増分h、始点の輝度値などの必要な情報が設定されると、次のステップS4で、図1のコードブック生成部9は、これら始点の輝度値と輝度の増分hとに基づいて、例えば線形補間の演算を行うことにより、各グループの輝度値を計算する。これにより、始点から終点に向かってグループ毎に徐々に輝度値が大きくなっていくパターンが生成される。

【0034】

本実施形態では、始点の輝度値を上記ステップS3のような処理によって与え

ているので、ブロック内の各グループに属する輝度値は全て、輝度値としてとり得る範囲の中間値よりも小さなものとなる。つまり、ここで生成されたベタの基本パターンのコードは、全体としての輝度値が中間値よりも小さい暗めの画像となっている。

【0035】

また、上記ステップS2では少なくとも1つ以上の増分hが与えられ、上記ステップS3では、それぞれの増分hの値をもとに始点の輝度値が複数与えられている。よって、図3(a)のような基本パターンのコードが、異なる輝度値で複数生成されていることになる。なお、増分hと始点の輝度値とをそれぞれ1つずつ与えることにより、ただ1つの基本パターンを生成するようにしても良い。

【0036】

以上のようにして生成された複数のコード(コードベクトル)は、ステップS5で、ベタの基本パターンのコードブックとしてコードブックデータメモリ10に格納される。

なお、この例では、ブロック内の始点から終点に向かって輝度値が徐々に大きくなるパターンを生成したが、これとは逆に、輝度値が徐々に小さくなるパターンを生成するようにしても良い。

【0037】

一方、エッジに関する基本パターンのコードとしては、輝度値の変化が急激なパターンとして、例えば図4に示すような12種類のパターンを作成することとした。ここで作成するエッジパターンは何れも、ブロック内の左辺を構成する4画素の少なくとも1つ以上にエッジ部分がかかっている。なお、ここでは12種類の基本パターンを示したが、この数に限定されるものではない。

【0038】

ここで、エッジの基本パターンを作成する手順を、図1に示したコードブック作成装置および図6に示すフローチャートを用いて説明する。

図6において、まずステップS11では、図1のパターン入力部5を用いて、エッジパターンとして採用すべき幾つかの基準パターンを入力する。ここでは、例えば原画像中の黒と白との差がはっきり現れているエッジにかかるブロックを

参考にして、その輝度構成をエッジパターンの輝度構成として入力する。

【0039】

以上のようにして原画像から幾つか（図4の例では12個）のパターンが入力されたら、次に、ステップS12で量子化部6は、それらのパターンを構成するブロック内の各画素の輝度値を量子化することにより、ブロック内の輝度値をある決まった値群のみで表現するようにする。

【0040】

次に、ステップS13で最小輝度値減算部7は、入力され量子化された各パターン毎に、そのブロック内の最小輝度値をブロック内の全画素の輝度値から減算する。これにより、そのブロック内の各画素の輝度値を、最小の輝度値に対する増分値（差分値）としてのみ表現する。そして、ステップS14で輝度値変更部8は、登録するエッジパターンにバリエーションを持たせるために、ブロック内の各画素の輝度値を変更したものも作成する。

【0041】

例えば、最小輝度値減算部7で生成されたパターンのブロック内全画素の輝度値を夫々 m 等分し、それぞれの等分値を各画素の輝度値として夫々設定することにより、パターンの数を m 倍に増やす。この処理が終了した時点で、生成された各パターンについて、そのブロック内の最小輝度値（ここではステップS13により0に設定されている）と最大輝度値との差、つまり増分 h' が分かる。

【0042】

次に、ステップS15では、図1の始点輝度値設定部4により、上記ブロック内の始点の輝度値（最小輝度値）を設定する。ここで始点の輝度値を与えるときは、例えば、

$0 \sim (\text{輝度値としてとり得る範囲の最大値} - \text{増分 } h')$

の間を k 等分し、それぞれの等分値を始点の輝度値として夫々設定する。なお、この場合も、図5のステップS3と同様に必ずしも等分値である必要はない。また、上記の範囲内でユーザが自分の判断で任意に入力するようにしても良い。

【0043】

このようにして始点の輝度値などの必要な情報が設定されると、次のステップ

S 16で、図1のコードブック生成部9は、これら始点の輝度値と上記ステップS 14で生成されたパターンの各画素の輝度値とに基づいて、ブロック内の各画素の輝度値を計算する。これにより、輝度値の変化が急激なエッジパターンのコードが複数生成される。このようにして生成された複数のコード（コードベクトル）は、ステップS 17で、エッジの基本パターンのコードブックとしてコードブックデータメモリ10に格納される。

【0044】

次に、上記のようにして作成され、コードブックデータメモリ10に記憶された基本パターンのコードブックを用いて、実際にベクトル量子化を行うための構成および動作について説明する。

図2は、本実施形態に係るベクトル量子化装置の概略構成を示した機能ブロック図である。また、図7は、このベクトル量子化装置の動作を示すフローチャートである。

【0045】

まず図7のステップS 21において、元画像入力部21は、圧縮対象とする任意の画像データを入力する。また、次のステップS 22で、コードブック演算部22は、コードブックデータメモリ10から記憶されている基本パターンのコードブックを読み込む。ここでは、ベタパターンおよびエッジパターンの全ての基本パターンを読み込む。

【0046】

基本パターンを読み込んだコードブック演算部22は、次のステップS 23で、読み込んだ基本パターンに対して、 90° の回転処理を4回行うことにより、基本パターンから異なるパターンのコードを作成する。例えば、ベタパターンとして図3(a)および(e)のパターンが基本パターンとして登録されていたとすると、この処理により、図3(b)～(d)、(f)～(h)のパターンが作成される。エッジパターンについても同様に、図4に示した12種類の基本パターンからこれらを回転させた異なるパターンが作成される。これにより、コードのパターン数は4倍に増える。

【0047】

コードブック演算部22はまた、次のステップS24で、上記ステップS23で得られた各パターンに対し、白黒を反転させる処理（輝度値を中間値で折り返す処理）を行うことにより、更に異なるパターンのコードを作成する。

このような回転処理および白黒反転処理を行うことにより、コードブックデータメモリ10に登録してあったパターン数がそれほど多くなくても、ベクトル量子化の際に実際に使用するパターン数は非常に多くなり、元画像と極めて類似するパターンが存在する可能性が非常に高くなる。

【0048】

ところで、ベタパターンの場合、図3から明らかなように、回転処理を行った場合に得られるパターンは、互いに重複することはない。また、あらかじめ作成されているベタの基本パターンは、全画素の輝度値が中間値より小さい暗めのパターンなので、白黒反転処理を行った場合に得られるパターンは、全画素の輝度値が中間値より大きい明るめのパターンとなる。よって、この場合も得られるパターンどうしが互いに重複することはない。

【0049】

一方、エッジパターンの場合は、最初の段階で原画像を参照して基準のパターンを入力する際に、回転処理や白黒反転処理をしたときに重複を生じないかどうかを想定して入力を行う。しかし、この作業は必ずしも容易ではない。そこで、例えば、図1のコードブック作成装置内に、回転処理や白黒反転処理を行い、その演算結果と元のデータとを比較して重複する場合にはエラーを出力するなどの演算部を設ければ、このような不都合を回避できる。

【0050】

次に、図7のステップS25では、上記ステップS21で入力された元画像データと、上記コードブック演算部22により生成された複数のコードデータとに基づいて、ベクトル量子化（VQ）の演算を実行する。すなわち、まず図2の類似度演算部23により、上記元画像データとコードデータとを用いて、ブロック毎に両者の類似度を算出する。

【0051】

類似度とは、元画像データから抽出されるブロック内の各画素値から成るベク

トルデータと、コードベクトルのブロック内の各画素値から成るベクトルデータとをある関数に入力し、どのくらい似ているかを数値化したものである。この関数の代表的なものとしては、2つの入力ベクトルデータのマンハッタン距離（差分絶対値距離）やユークリッド距離を求める関数が挙げられる。

【0052】

さらに、コード決定部24は、各ブロック毎に、上記コードブック演算部22により生成された複数のコードベクトルの中から、類似度が最も大きい（マンハッタン距離あるいはユークリッド距離が最も小さい）コードベクトルを夫々決定する。そして、ステップS26で、上記決定したコードベクトルに対応するコードを当該ブロックに当てはめて、圧縮データとして出力する。

【0053】

以上詳しく説明したように、本実施形態によれば、あらゆる種類の画像はベタパターンとエッジパターンとの組み合わせから構成されることに着目し、ベタパターンのコードとエッジパターンのコードとをあらかじめ作成するようにしている。その際、図5および図6に示したように、入力された基準の情報をもとに、あらかじめ決められた処理手順によってそれぞれの基本パターンを標準化して作成するようにしている。

【0054】

したがって、このようにして作成されたベタパターンのコードとエッジパターンのコードとを用いることにより、輝度値が単調に変化するデータや急激に変化するデータを含む種々のデータに対してデータ圧縮を実施した場合に、その圧縮データから再生される画像の画質を向上させることができる。すなわち、本実施形態によれば、種々の画像について、高品位の再生画像を得ることが可能な汎用性の高いコードブックを実現することができる。

【0055】

また、本実施形態では、上述のコードブックを用いて実際にベクトル量子化を行う際に、回転処理や白黒反転処理によって異なるバリエーションのコードを生成しているので、元画像とより近いパターンのコードを当てはめるようにすることができ、再生画像の品質を更に向上させることができる。また、この場合、こ

これらの生成されるコードも含めてあらかじめコードブックデータメモリ10に記憶しておかなくても良いので、コードブックデータメモリ10の容量を小さく抑えることができる。

【0056】

なお、以上の実施形態では、基本パターンのコードブックをあらかじめ作成して、これをコードブックデータメモリ10に記憶するようにしているが、この基本パターンを生成するための基準となる情報（例えば、始点の位置、始点輝度値、始点から終点までの増分などの情報）だけを記憶しておき、実際のベクトル量子化の実行時に基本パターンも含めて様々なパターンのコードを作成するようにしても良い。このようにした場合は、メモリ容量を更に小さく抑えることができる。

【0057】

また、以上の実施形態では、回転処理や白黒反転処理をベクトル量子化の実行時に行っているが、ベクトル量子化の実行に先立ってあらかじめ行っておき、コードブックデータメモリ10に記憶するようにしても良い。この場合は、コードブックデータメモリ10の容量は大きくなるが、汎用的なコードブックを提供できるという利点は有する。また、ベクトル量子化の実行時における演算量を減らせるので、処理が高速になる。

【0058】

また、上述の実施形態では、利用できるコードベクトルのパターン数を多くするために、図5のステップS3および図6のステップS15において、始点の輝度値として複数の等分値を採用している。これに対し、コードブックデータメモリ10の容量をできるだけ小さく抑えたい場合は、上記した範囲内で1つの輝度値のみを採用するようにしても良い。

【0059】

また、図6のステップS13～S16の処理も、利用できるコードベクトルのパターン数を多くするための処理である。よって、上記と同様にコードブックデータメモリ10の容量をできるだけ小さく抑えたい場合は、これらのステップは省略し、ステップS12で画素値を標準化する処理を行うだけにしても良い。

ただし、コードブックデータメモリ 10 の容量を小さくするだけでなく、再生画像の画質もできるだけ向上させるためには、上記した実施形態のように処理するのが望ましい。

【0060】

また、以上の実施形態では、回転処理と白黒反転処理の双方を行っているが、何れか一方のみ行うようにしても良い。

さらに、以上の実施形態では、類似度演算部 23 で類似度を算出する際に、コードブック演算部 22 によってコードベクトルを回転、反転させる処理を行っている。これとは逆に、コードベクトルはそのままに、元画像の方を回転、反転させて類似度を算出するようにしても良い。

【0061】

また、以上の実施形態では、ベタの基本パターンを作成する際に、全体として暗めのパターンを作成しているが、これとは逆に、明るめのパターンを作成するようにしても良い。

【0062】

上記図 1 および図 2 に示した各機能ブロックは、例えば CPU または MPU、ROM および RAM 等からなるマイクロコンピュータシステムによって構成し、その動作を ROM や RAM に格納された作業プログラムに従って実現するようにしても良いし、ハードウェア的に構成してもよい。この場合、上記の例では図 1 と図 2 を別個に図示していたが、これらの構成を合わせて 1 つのベクトル量子化装置としても良い。

【0063】

また、上記各機能ブロックの機能を実現するように当該機能を実現するための作業プログラムコードを外部の記録媒体からコンピュータに供給し、そのプログラムに従って上記各機能ブロックを動作させるようにしても良い。この場合、かかるプログラムを記憶する記録媒体としては、例えばフロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-I、CD-R、CD-RW、DVD、zip、磁気テープ、あるいは不揮発性のメモリカード等を用いることができる。

【0064】

【発明の効果】

本発明は上述したように、ブロック内でデータ値が徐々に変化していくベタパターンコードと、ブロック内でデータ値が急激に変化するエッジパターンコードとを各々基本パターンとして作成するようにしたので、ベタパターンとエッジパターンとの組み合わせから構成されるあらゆる種類の画像に対応することができる。

また、本発明の他の特徴によれば、上記の基本パターンをあらかじめ作成して記憶しておき、ベクトル量子化の実行の際に、記憶されている各々のパターンコードに対して演算を施すことによってこの基本パターンとは異なるパターンコードを作成するようにしたので、ベクトル量子化を行う際に演算によって様々なパターンコードが生成され、元画像とより近いパターンコードを当てはめるようにすることができ、再生画像の品質を向上させることができる。また、記憶手段には基本パターンのみを記憶しておけば良いので、メモリ容量を小さく抑えることもできる。

以上のことから、本発明によれば、種々の画像に対応できる汎用性の高いコードブックを、これを記憶しておくためのメモリ容量を増大させることなく実現することができる。

【図面の簡単な説明】

【図1】

本発明の一実施形態に係るコードブック作成装置の構成例を示すブロック図である。

【図2】

本発明の一実施形態に係るベクトル量子化装置の構成例を示すブロック図である。

【図3】

作成されるベタパターンのコードの例を示す図である。

【図4】

作成されるエッジの基本パターンのコードの例を示す図である。

【図5】

ベタの基本パターンを作成する際の手順を示すフローチャートである。

【図6】

エッジの基本パターンを作成する際の手順を示すフローチャートである。

【図7】

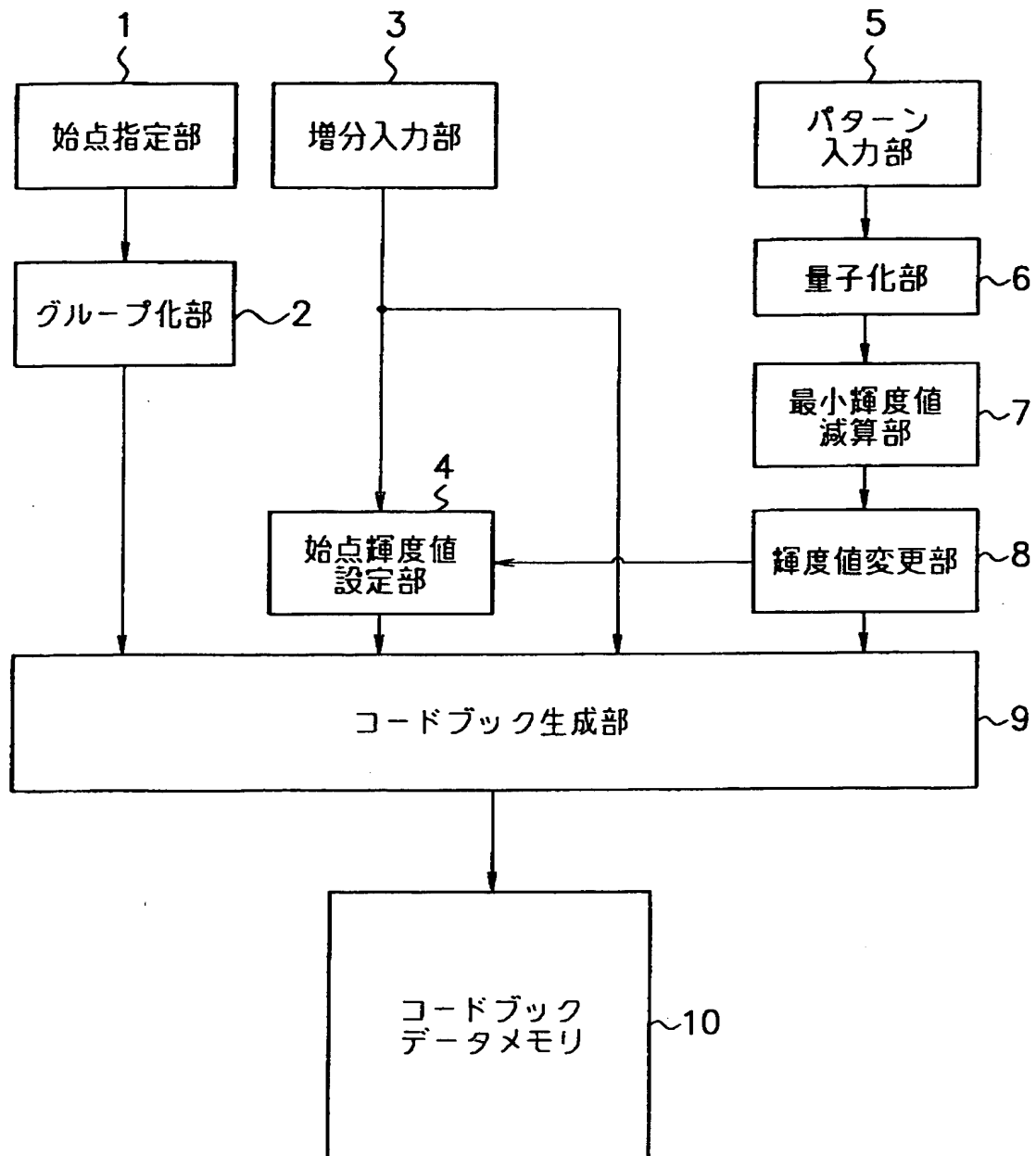
ベクトル量子化を実施する際の手順を示すフローチャートである。

【符号の説明】

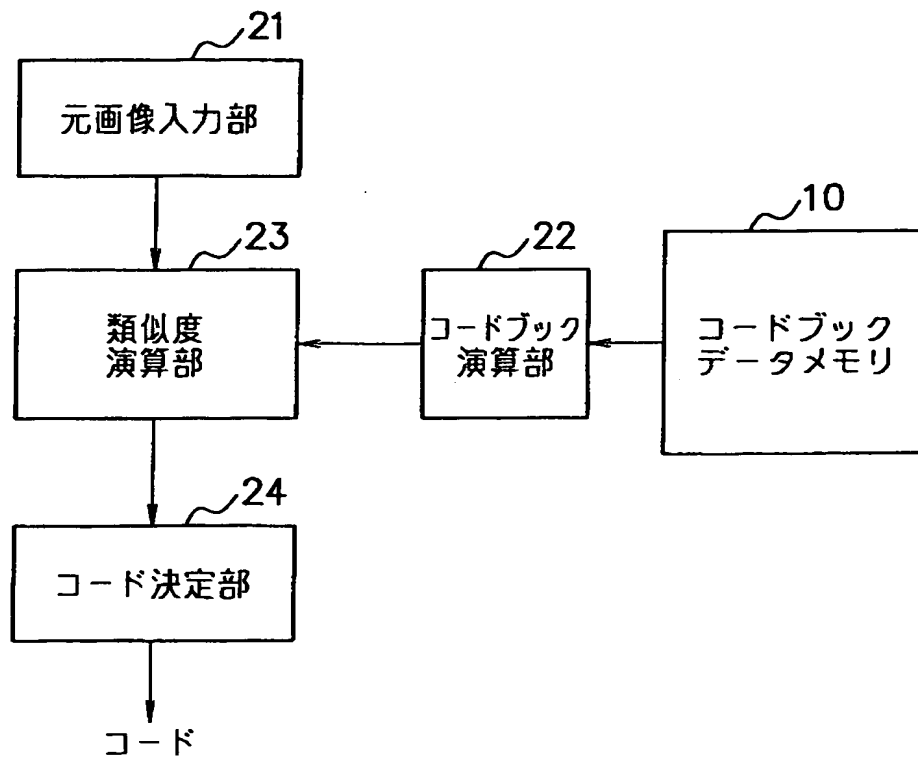
- 1 始点指定部
- 2 グループ化部
- 3 増分入力部
- 4 始点輝度値設定部
- 5 パターン入力部
- 6 量子化部
- 7 最小輝度値減算部
- 8 輝度値変更部
- 9 コードブック生成部
- 10 コードブックデータメモリ
- 21 元画像入力部
- 22 コードブック演算部
- 23 類似度演算部
- 24 コード決定部

【書類名】 図面

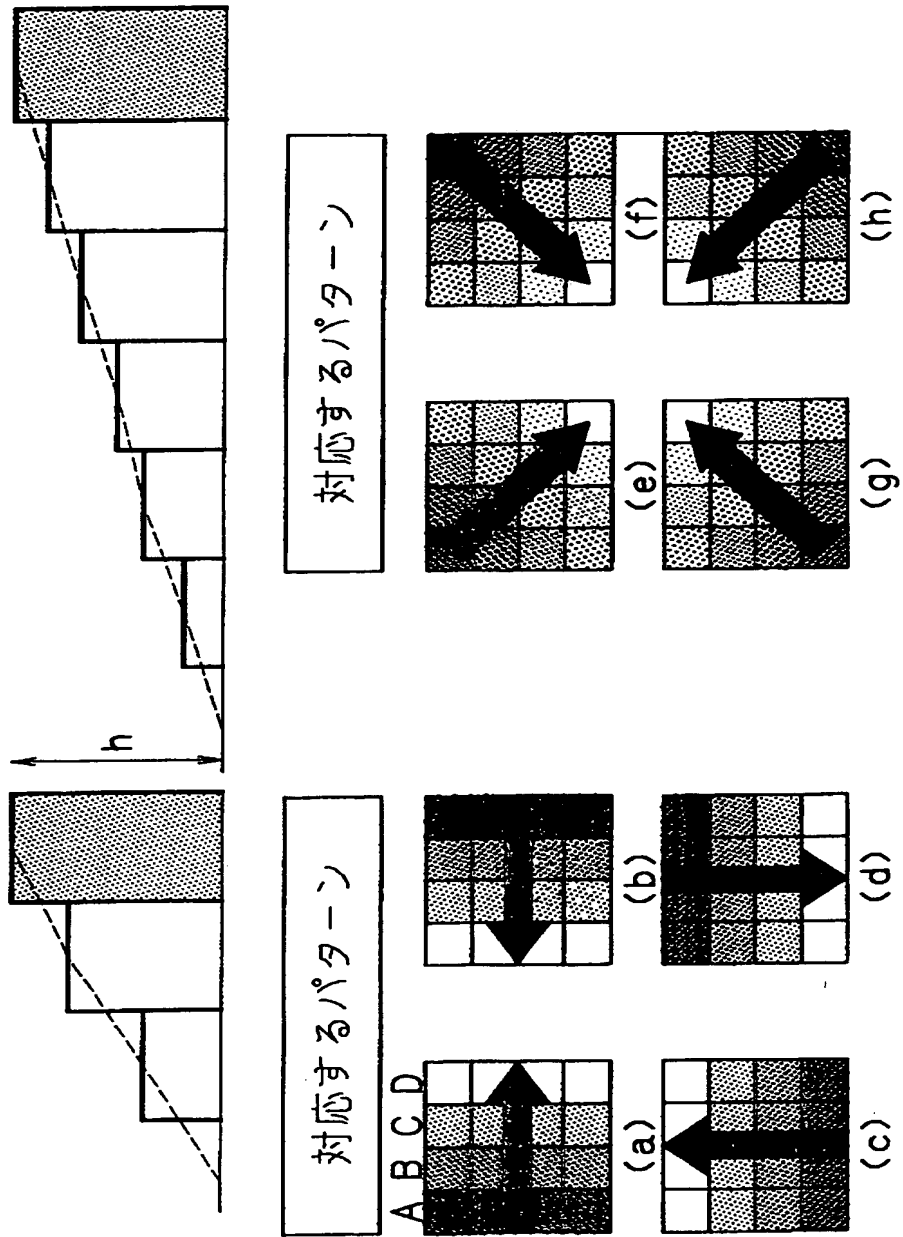
【図 1】



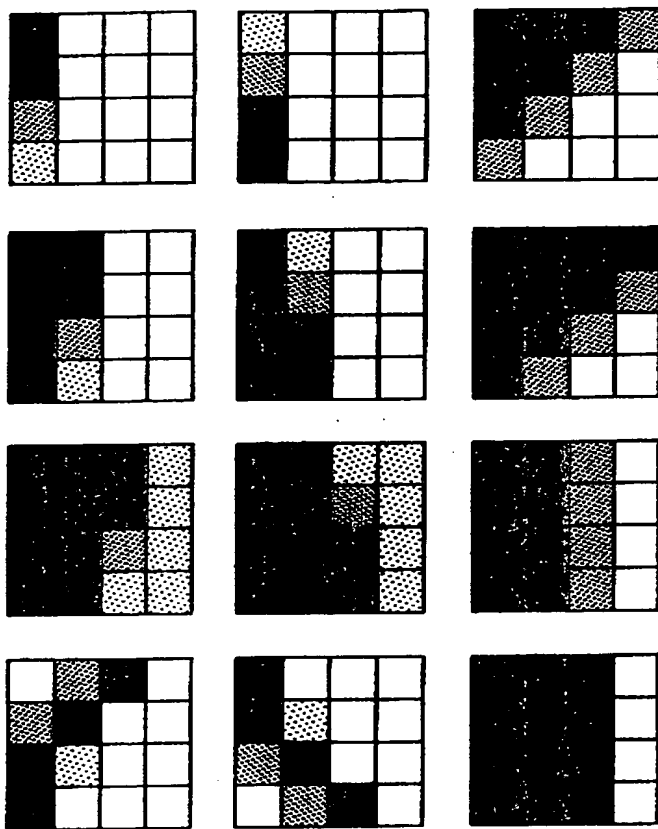
【図 2】



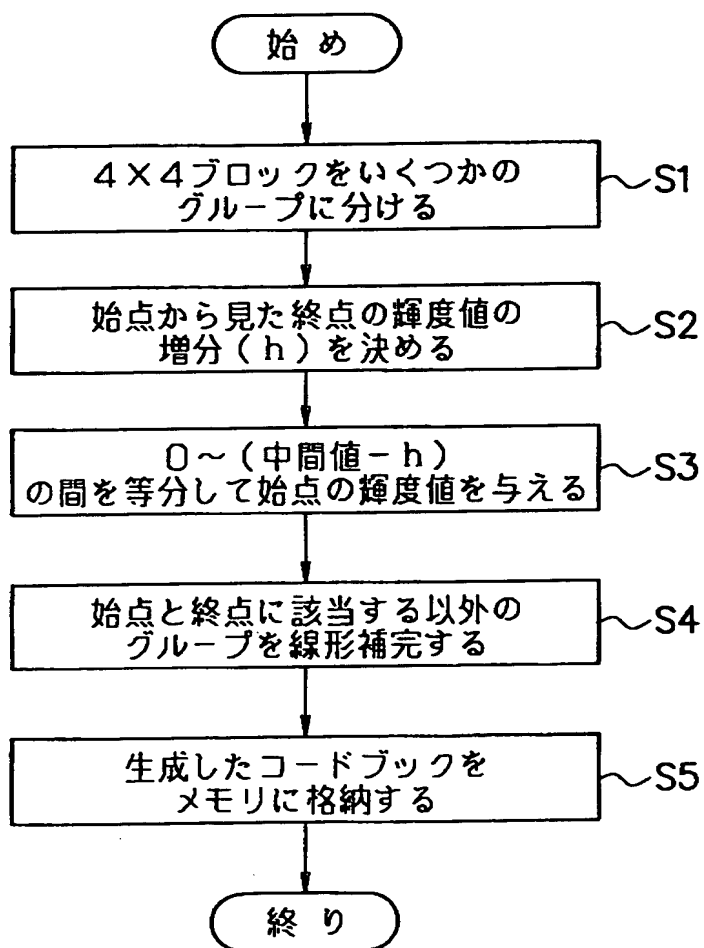
【図 3】



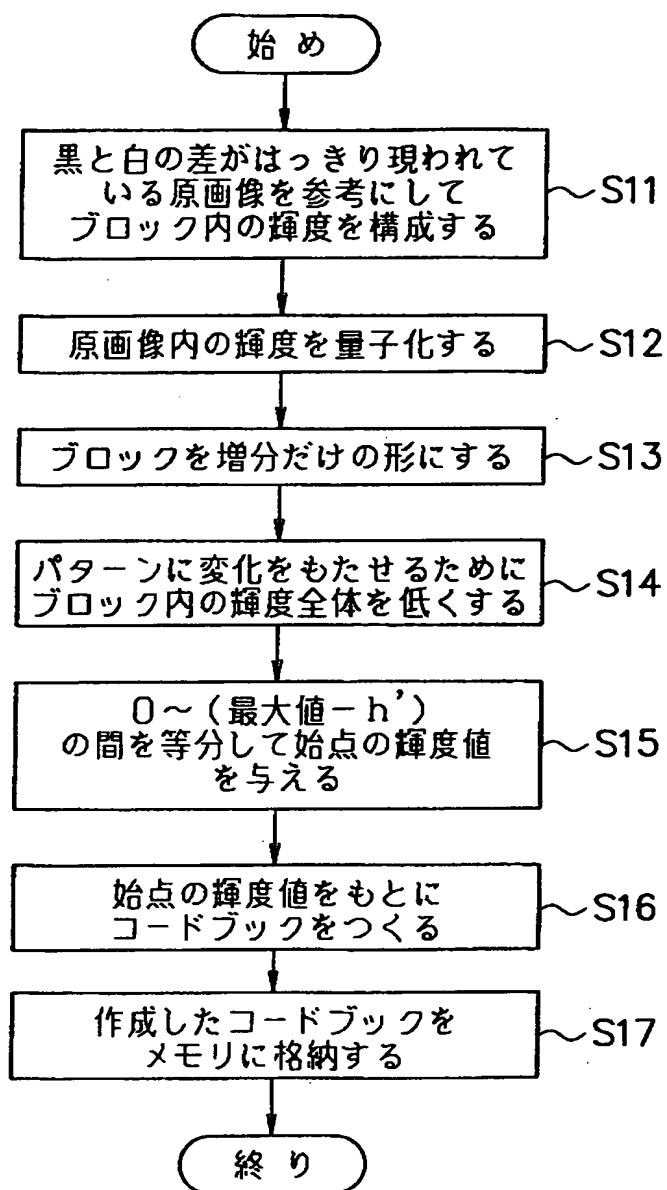
【図4】



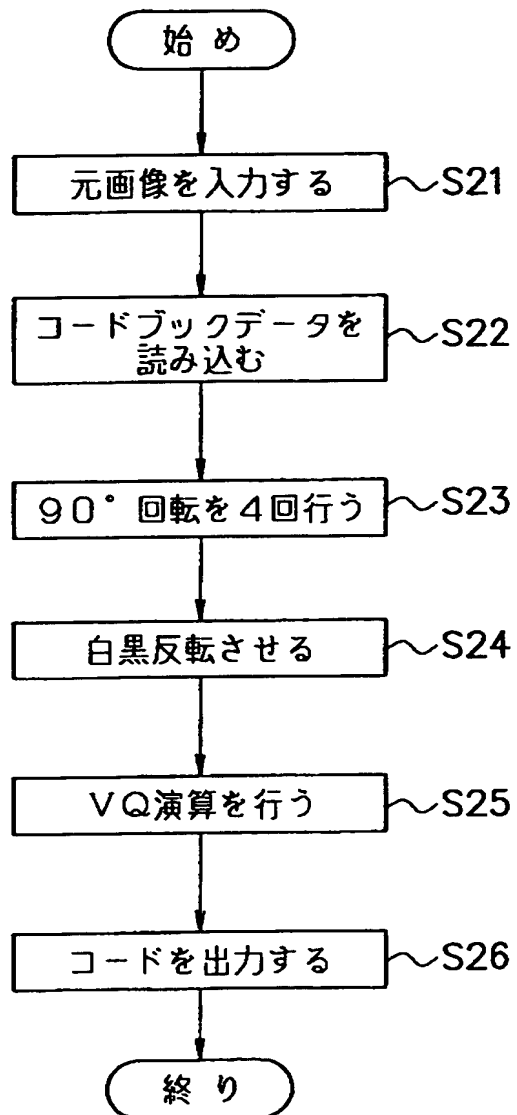
【図 5】



【図 6】



【図 7】



【書類名】 要約書

【要約】

【課題】 種々の画像に対応できる汎用性の高いコードブックを、これを記憶しておくためのメモリ容量を増大させることなく実現できるようにする。

【解決手段】 ブロック内で輝度値が徐々に変化していくベタパターンのコードを作成するとともに、輝度値が急激に変化するエッジパターンのコードを作成し、得られたコードをメモリ 10 に記憶しておく。そして、ベクトル量子化を実際に実行する際に、メモリ 10 に記憶されている各々のパターンコードに対して演算を施すことにより、これらとは異なるパターンのコードを作成し、ベクトル量子化に利用するようにすることにより、ベタパターンとエッジパターンとで構成されるあらゆる種類の画像に対応できるようにするとともに、メモリ 10 には基本パターンのみを記憶しておけば良いようにして、メモリ容量を小さく抑えることができるようにする。

【選択図】 図 1

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】
【識別番号】 000205041
【住所又は居所】 宮城県仙台市青葉区米ヶ袋 2-1-17-301
【氏名又は名称】 大見 忠弘
【代理人】 申請人
【識別番号】 100090273
【住所又は居所】 東京都豊島区東池袋 1丁目 17番 8号 池袋TGホ
ーメストビル 5階 國分特許事務所
【氏名又は名称】 國分 孝悦

出 願 人 履 歴 情 報

識別番号

[000205041]

1. 変更年月日 1990年 8月27日

[変更理由] 新規登録

住 所 宮城県仙台市青葉区米ヶ袋2-1-17-301

氏 名 大見 忠弘

THIS PAGE BLANK (USPTO)